PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L 9/30

A1

(11) International Publication Number: WO 99/16209

(43) International Publication Date: 1 April 1999 (01.04.99)

(21) International Application Number: PCT/US98/20036

(22) International Filing Date: 22 September 1998 (22.09.98)

(30) Priority Data:
60/059,455 22 September 1997 (22.09.97) US

(71) Applicant: EOLAS TECHNOLOGIES, INCORPORATED [US/US]; 10 East Ontario Street, Chicago, IL 60611 (US).

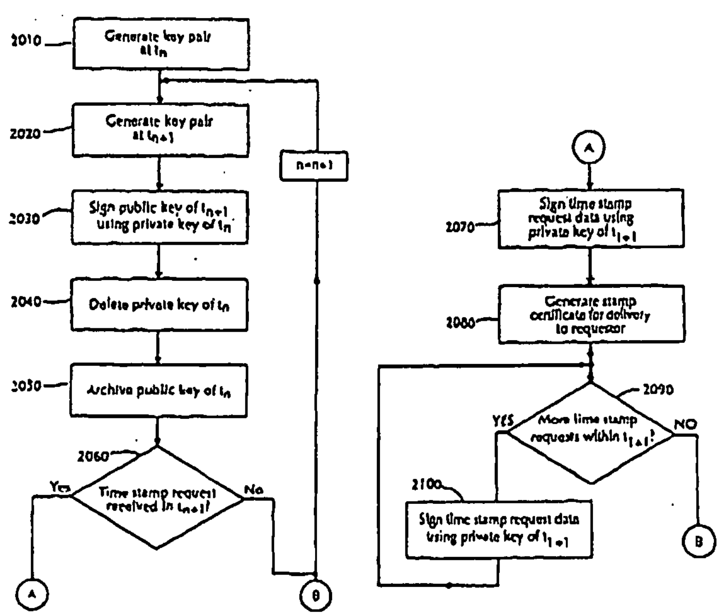
(72) Inventor: DOYLE, Michael, D.; 824 Dawes Avenue, Wheaton, IL 60187 (US).

(74) Agent: KOLEFAS, Chris; Baker & McKenzie, 805 Third Avenue, New York, NY 10022 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND SYSTEM FOR TRANSIENT KEY DIGITAL TIME STAMPS



(57) Abstract

Irrefutable public key digital signature time-stamps (1040) are created and used based upon, for example, the concept of transient time-interval-related secret cryptographic keys (2010), which are used to digitally sign (2030) submitted data during specific time intervals, and then permanently destroyed (2040). The public-key correlate for each time interval is saved for future authentication of the content of time-stamped data and time of creation of time-stamped data. The validity of the public keys is ensured through the certification of each time interval's public key using the previous time interval's secret key, immediately before that secret key is destroyed.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND SYSTEM FOR TRANSIENT KEY DIGITAL TIME STAMPS

FIELD OF THE INVENTION

The present invention relates to a method for digital time stamping data. More particularly, the present invention relates to the digital time stamping of data, without the need for subsequent third party verification, by the chaining of key pairs, the key pairs being generated for particular time intervals.

BACKGROUND INFORMATION

The concept of chain of evidence has long been a fundamental tenet of the U.S. judicial system. Many legal situations depend upon the ability to prove that a certain piece of evidence existed at a certain point in time and that it has not been subsequently altered. In the past, when most of the possible types of evidence consisted of material objects, there was a need for a protocol of a "chain of witnesses" to testify to the veracity of an evidentiary object in question. Historically, if the evidence was under the control of only a finite set of individuals, and if all of those individuals could testify as to the location and state of the object, then the court would accept the claim of authenticity of the evidence.

Of course, such a system is dependent upon the availability of trustworthy witnesses that will be available and willing to testify in any given circumstance. Often times, witnesses are available, but not trustworthy, or vice versa. This is particularly the case with respect to document authentication,

where the details of when a specific document was created or signed is in question. Clearly, a system was needed to allow one to easily obtain a "witness on demand" in many situations.

This concept of evidentiary authentication is so important to so many areas of endeavor, that a formalized system of professional document witnesses was developed, for example, called the Notary Public service. Notary Publics would, for a fee, attest to such things as the existence of a document and the identity of the document holder or signer. Of course a notary could not swear to any knowledge of the actual contents of a document, since that would have required that the notary keep copies, in perpetuity, of every document ever witnessed -- an impractical requirement. Much of the trust held in the notary public system related to a generally-held belief that it was impossible or impractical to forge a notary public's stamp and signature, or to buy a notary public's testimony. As computer graphics and desktop publishing technology advances, however, the level of difficulty of creating forged documents and signatures decreases significantly. A result of this technological advance is the fact that some states, such as California, no longer accept notarization as absolute proof of document validity.

As more and more of the information of import in personal and business transactions becomes digital in form, the usefulness of notary-public-style authentication mechanisms decreases. Much of this information is stored, accessed and managed through computer database management systems. All major database systems permit time stamping of data in records. Many commercial and governmental systems depend upon the assumption of veracity of such database time stamps. The presumption is that, if the organization is trustworthy, then

the time stamps in their databases can be believed. In practice, this assertion requires a large degree of, to borrow a literary term, "willing suspension of disbelief." No one, of course, can safely assume that all individuals within a large organization are trustworthy, even if the organization, itself, is believed to be so. Furthermore, it is now well known that no conventional computer database system is immune from the possibility of data tampering or "hacking" by dishonest individuals.

One approach that has been developed to deal with some of this problem is based upon a technology called "public key" cryptography. One of the most well known of this type of system is the program called Pretty Good Privacy, distributed by the Massachusetts Institute of Technology, which makes use of the Rivest-Shamir-Adleman (RSA) public key cryptosystem. Such systems are built around the concept of encrypting data in such a way that allows both secure transmission and authentication of sensitive data. Public key systems employ a pair of cryptographic keys for each encryption/decryption event. One key is kept secret by the owner (e.g., the private key), and the other key is publicly distributed (e.g., the public key). A message encrypted with one of the keys in a key pair can only be decrypted with the other key, and vice versa.

This system allows, for example, the encryption of data by one individual, using a second individual's public key. The message could then be sent to a second individual over unsecure channels, and only the second individual could access the unencrypted data, since it could only be decrypted with the second individual's private key.

Prior to using the second individual's public key to encrypt the data, the first individual could have used his or her private key to encrypt the data, thereby digitally "signing" the data. The recipient could then use the sender's public key to decrypt it, thus proving that it actually came from the sender, since only the sender could have used the correct secret key to sign the data. Such a system provides both confidentiality of data and a mechanism for authentication of the identity of the sender. It also proves that the data could not have been altered in any way since the time it was encrypted by the sender. Public keys, themselves, can be "certified" by signing them with a trusted individual's secret key (e.g., a digital signature). Others can then assess the authenticity of published public keys by authenticating them using that trusted individual's public key. If that trusted individual later loses faith in the validity of the certified key, then he can issue a so-called revocation certificate, signed by the trusted individual's private key, that notifies others that the previously-certified public key is no longer to be trusted in the future.

Public key algorithms are notoriously slow. For this reason, virtually all public key digital signature systems use what is called a "cryptographically-strong one-way hash function" to create what is called a "message digest" from the data to be signed. This message digest is a unique representation of that data, sort of a data fingerprint, that is typically much smaller than the original data. For example, the message digests that PGP uses are only 128 bits in length. The message digest is then encrypted using the sender's secret key before sending the data to the recipient. The recipient can then use the sender's public key to automatically decrypt the message digest and then verify that it does indeed match the original data. This is a very secure system, since it is

computationally infeasible for an attacker to devise a substitute message that would provide an identical message digest. Most estimates state that it would take 10^12 or more years (taking into account Gordon Moore's "law" relating to increases in chip capacity over time) to successfully fake a 128-bit message digest using the algorithm employs by the PGP software package. Also, changing even a single byte of a digested message would cause the hash function to be unable to match the message digest to the unencrypted data.

Public key digital signatures, therefore, can irrefutably prove that signed data was originally signed by a given secret key and that the data has not changed in any way since the signature was made. Systems such as PGP routinely attach time-stamps to both key pairs at their creation, and to digital signatures, each time they are created. Such time-stamps, however, are dependent only upon the internal clocks within the computers being used, and thus are subject to inaccuracies or falsification by, for example, an individual intentionally changing the time on a computer's clock in order to make it falsely appear that a given digital signature was created at a specific point in time.

For this reason, a new type of notary public has arisen, which uses public-key digital signatures to notarize, for a fee, digital information typically submitted over the Internet. These so-called "digital notaries" are, essentially, businesses that provide such a service and agree to attest to the veracity of both the content of the original data, as well as the time at which the signature was made. This is a major improvement over the notary public concept of old, since the new digital notary services can testify to the fact that data which has been digitally signed by their service existed at a

certain point in time, and that it hasn't been altered in any way since that point in time. The largest problem with such digital notary services, and also a motivating reason for the method according to the present invention, is the fact that the authenticity of such digital-notary-generated digital signatures is wholly dependent upon the trustworthiness of the institution and individuals running the digital notary service.

To solve this problem, a system is needed that will automatically and rigorously prove the veracity of digital signature time-stamps, without depending upon the trustworthiness of the institution or individuals administering a digital notary service. Transient-key digital time-stamps according to an embodiment of the present invention provide these capabilities.

SUMMARY OF THE INVENTION

According to an embodiment of the present invention, irrefutable public key digital signature time-stamps are created and used. The system is based upon, for example, the concept of transient time-interval-related secret cryptographic keys, which are used to digitally sign submitted data during specific time intervals, and then are permanently destroyed. The public-key correlate for each time interval is saved for future authentication of the content of time-stamped data and time of creation of time-stamped data. The validity of the public keys is ensured through the certification (e.g., signing) of each time interval's public key using the previous time interval's secret key, immediately before that secret key is destroyed.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates an exemplary flowchart for a method for digital time stamping according to an embodiment of the present invention.

Figure 2A illustrates a portion of an exemplary flowchart for another method for digital time stamping according to an embodiment of the present invention.

Figure 2B illustrates another portion of an exemplary flowchart for another method for digital time stamping according to an embodiment of the present invention.

Figure 3A illustrates a first exemplary embodiment for a time stamping system according to the present invention.

Figure 3B illustrates a second exemplary embodiment for a time stamping system according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The digital time stamping method according to an embodiment of the present invention provides a mechanism to irrefutably prove that a collection of data existed at a given interval of time and has not changed since that interval of time. A significant advantage of the present invention is that it provides non-repudiation to the user. It is difficult to deny the veracity of the time-stamp certificates generated by the method according to an embodiment of the present invention. For example, the system does not depend upon the trustworthiness (or later existence) of any external

"certification authority" or any external time tracking system. Rather, all that is needed to authenticate the time stamp generated according to an embodiment of the present invention is, for example, the time-stamped data, the signature from the time-stamp certificate, the time interval's public key from the time-stamp certificate, and a standard public-key authentication program, such as either the free or commercial version of PGP. Other public key encryption programs such as the J/CRYPTO Professional Cryptography Classes for Java Developers (http://www.baltimore.ie/products/jcrypto/index.html) could also be used with the present invention. Moreover, the method according to an embodiment of the present invention will work with any kind of computer data.

Systems utilizing the method for transient-key digital time stamps according to the present invention can be set up as, for example, Internet servers, stamping all requests on a fee-for-service basis. The time of creation and the internal state of information can be proven without endangering the confidentiality of sensitive data. The time stamping method according to an embodiment of the present invention makes the method suitable for use in, for example, invention documentation systems. Accordingly, the method according to an embodiment of the present invention can also be used to authenticate critical confidential records, such as medical records and financial transactions, can be easily adapted to any computing platform, and is not dependent upon any specific public-key algorithm.

Figure 1 illustrates an exemplary flowchart for a digital time stamping method according to an embodiment of the present invention. In step 1010 a key pair is generated at time

interval t_n. As is known in the art, the key pair includes a public key and a private key. The time interval can be any defined period, e.g., every second, 10 seconds, minute or 10 minutes. The current time interval is referred to as t_n. In step 1020, it is determined if a time stamp request is received during time interval t_n. If no time stamp request is received during time interval t_n, then the process returns to step 1010 to generate a new key pair for the next time interval, n being incremented by 1 to indicate the next time interval.

If a time stamp request is received during time interval t_n, in step 1030 the data accompanying the time stamp request is automatically signed. For example, a conventional message digest for the data could be generated that would be automatically encrypted using the private key of time interval t_n. As a result of signing the data, the signature of the time stamp can only be decrypted using the public key of time interval t_n. In step 1040, a time stamp certificate is generated for delivery to the requestor indicating the temporal existence of the data. In step 1050, it is determined if additional time stamp requests are received within time interval t_n.

If no additional time stamp requests are received, then the private key for time interval t_n is deleted in step 1060 and the process returns to step 1010 to generate a key pair for the next time interval, n being incremented by 1. If further time stamp requests are received during time interval t_n, then the process returns to step 1030 to process each further time stamp request. As indicated in step 1060, the private key for time interval t_n is deleted at the end of the time interval and the public key would be, for example, archived for subsequent

use to decrypt the time stamp. Thus, a separate private key is used to automatically time stamp the data associated with a time stamp request received during each defined time interval according to an embodiment of the present invention.

The process according to an embodiment of the present invention illustrated in Figure 1 differs from prior art systems in that, for example, the key pairs are automatically generated every defined time interval and the data accompanying the time stamp request is automatically signed using the private key of the time interval that the time stamp request is received, the private key being deleted after the time interval. In contrast, prior art time stamping systems would use a single private key to sign all time stamp requests and employ a separate mechanism, usually based on the computer system implementing the time stamp, to provide the time stamp data. Also unlike the time stamping method according to an embodiment of the present invention, some prior art systems would chain together the message digests for sequentially-submitted documents that have been signed to generate the message digest encrypted for the time stamp, for example, as described in U.S. Patent No. 5,136,647, which is hereby incorporated by reference.

Figure 2A illustrates an exemplary flowchart for a digital time stamping method according to another embodiment of the present invention. In step 2010 a key pair is generated. As is known in the art, the key pair includes a public key and a private key. According to an embodiment of the present invention, a key pair is generated for each time interval utilized by the system implementing the time stamping method. The implementing system can include, for example, a conventional general purpose computer, such as a

microprocessor based personal computer or server. In an embodiment of the present invention, the method is implemented in software that executes on a client-server computer system architecture. The time interval can be any defined period, e.g., every second, 10 seconds, minute or 10 minutes. The current time interval is referred to as t_n and the next time interval is referred to as t_{n-1}. For the purposes of time stamping documents, accuracy to the minute may be sufficient for subsequent authentication purposes.

In step 2020, another key pair is generated at time t_{n-1}. Like the first key pair, the next key pair also has a public key and a private key. To generate the key pairs in steps 2010 and 2020, a conventional digital time stamping system such as PGP could be modified to automatically generate key pairs every defined time interval. For example, conventional digital time stamping systems are designed for users to generate key pairs, usually via user I/O with the system to input the information necessary to generate a key pair (e.g., a pass phrase and a random seed required by PGP). According to an embodiment of the present invention, the source code for such systems could be modified to generate, for example, a pass phrase and a random seed that would be automatically fed to the key pair generation algorithm for each defined time interval, thereby automatically providing the input normally provided by a user to generate a key pair.

In step 2030, the public key of time interval t_{n-1} is signed using the private key of time interval t_n. For example, a conventional message digest for the public key of time interval t_{n-1} could be generated that would be encrypted using the private key of time interval t_n. As a result of signing the public key of time interval t_{n-1}, the signature of the public key can only be decrypted using the public key of time

interval t_n. The signing of the public key of time interval t_{n+1} using the private key of time interval t_n could be accomplished, for example, using script based control of existing software, such as the PGP software (e.g., a single command line instructing that one key sign another key). In step 2040, the private key of time interval t_n is deleted. Thus, the private key for time interval t_n exists for the duration of time interval t_n and for the time necessary during time interval t_{n-1} to sign the public key of time interval t_{n-1}. In step 2050, the public key for time interval t_n is archived for subsequent use, e.g., to decrypt the time stamp on the public key of time interval t_{n+1}.

In step 2060, it is determined if a time stamp request is received during time interval t_{n-1}. If no time stamp request is received, then the process returns to step 2020 to generate a key pair for the next time interval, n being incremented by 1. If a time stamp request is received during time interval t_{n-1}, in step 2070, illustrated in Figure 2B, the data accompanying the time stamp request is signed using the private key of time interval t_{n-1}. For example, as is known in the art, a conventional message digest for the data to be time stamped according to an embodiment of the present invention could be generated that would be encrypted using the private key of time interval t_{n-1}. As a result of signing the data using the private key of time interval t_{n-1}, the signature of the time stamp could only be decrypted using the public key of time interval t_{n-1}, which public key itself has been time stamped according to an embodiment of the present invention and can be authenticated only using the public key of the prior time interval t_n as described above. Therefore, using an embodiment of the method according to the present invention, authentication of the time stamp on data is self-validated as the keys for two time intervals have been chained together.

No independent third party is required to verify that the time stamp on the data is accurate. In another exemplary embodiment, the key pair for t_{n+1} is generated and certified in advance, during the end of the prior time interval t_n, to insure that the key pair for time interval t_{n+1} is available immediately at the beginning of t_{n+1}.

In step 2080, a stamp certificate is generated for delivery to the requesting party. According to an exemplary embodiment of the present invention, such a stamp certificate includes a digital signature of the submitted data and the certified public keys for time intervals t_n and t_{n-1}. In step 2090, it is determined if any additional time stamp requests are received within time interval t_{n-1}. If no further time stamp requests are received within time interval t_{n+1}, the process returns to point B on Figure 2A to generate the key pair for the next time interval. If another time stamp request is received during time interval t_{n+1}, in step 2100 the data accompanying the time stamp request is signed using the private key of time interval t_{n+1} as described above and the process loops back to step 2090 until no further time stamp requests are received during time interval t_{n-1}.

The method according to an embodiment of the present invention for time stamping data can be implemented, for example as software, firmware or hard-wired logic using a suitable general purpose computer. For example, the software implementation of the present invention can be written in the Java programming language, that can run on any platform.

Figure 3A illustrates an exemplary client-server architecture

for implementing the time stamping method according to an embodiment of the present invention. In a client-server architecture, the server portion of time stamping program for an embodiment of the present invention would reside in, for example, a memory 3015 of the server 3010. The time stamping program would execute on the cpu 3016 connected to the memory 3015. The server 3010 is connected to the client 3020 via, for example, a connection 3030, such as a LAN, WAN or Internet connection. The client computer 3020 would include a time stamping client portion of the method according to an embodiment of the present invention residing in a memory 3025, the time stamping client program executing on the cpu 3026 connected to the memory 3025. An I/O device 3040, such as a keyboard or mouse provides user access to the time stamping method according to an embodiment of the present invention.

In operation, for example, a user would identify data to be time stamped via the I/O device 3040 which would cause the client application program stored in memory 3025 to execute in memory 3026 and generate a message digest for the data, for example in a manner known in the art. The message digest would be transmitted via connection 3030 to server 3010, where the application program stored in memory 3015 would execute in memory 3016 to time stamp the message digest and return a time stamp certificate to client computer 3020 via connection 3030, for example as described in Figures 1 or 2A-2B.

In an alternative implementation of the client-server architecture illustrated in Figure 3A, the signing could occur at the client computer 3020. For example, via the I/O device 3040, a user could identify data to be time stamped and submit the stamp request to the server computer 3010 via connection 3030 without providing a message digest for the data. In

response to the stamp request, the server 3010 would generate a key pair for the current time interval according to an embodiment of the present invention (e.g., with a public key signed by the private key of the prior time interval key pair) and return the key pair for the current time interval, the passphrase for the time interval's private key, and the public key from the prior time interval to the client computer 3020. To ensure the secrecy of the transmission from the server 3010 to the client 3020, the connection 3030 can include, for example, a secure channel using SECURE SOCKETS LAYER (SSL). Once the client 3020 receives the transmission from the server 3010, the client can generate the message digest and sign the message digest of the time stamp request using the private key of the current time interval, for example in a manner as is known in the art. After the time stamp is created, the client-side copies of the associated private key and passphrase are then immediately deleted.

In yet another alternative embodiment of the client-server architecture illustrated in Figure 3A, the client computer 3020 can generate its own key pair and use a key pair generated by the server 3010 to time stamp the public key of the key pair generated by the client computer 3020. For example, the client computer 3020 would generate a key pair and transmit the public key of the key pair to the server 3010 via connection 3030. The private key of a key pair generated by the server 3010 for the current time interval would be used to sign the public key from the client 3020. The signed public key and the public key of the key pair generated by the server would be transmitted back to the client 3020. The private key from the key pair generated by the client 3020 would be used to time stamp the data. Immediately after the time stamp was produced, the client-side private key would be immediately deleted, then the client-side public key would be

revoked by using the server-side private key to issue a revocation certificate for the client-side public key. The private key from the server 3010 would be destroyed. The revocation certificate would be incorporated into the time stamp certificate, together with the signature of the data, the server-side public keys for the current and previous time intervals, and the client-side public key.

Figure 3B illustrates an alternative embodiment for a system implementing the time stamping method according to an embodiment of the present invention. In Figure 3B, the time stamping method is carried out in a single computer system 3100, such as a relational database system or a financial transaction system. Computer system 3100 includes a memory 3115 connected to a cpu 3116. An I/O device 3140, such as a keyboard or mouse, is connected to the computer 3100 and provides user access to the time stamping method according to an embodiment of the present invention. The memory 3115 would contain, for example, both the resident program to generate the message digests for data to be time stamped and the time stamping program according to an embodiment of the present invention.

According to the illustrative embodiment of Figure 3B, either the user would identify data to be time stamped via the I/O device 3140 or the system would automatically identify data to be time stamped, for example as in response to a database transaction. Identification of the data to be time stamped then would cause the resident program stored in memory 3115 to execute in CPU 3116 and generate a message digest for the data. The message digest would be provided to the application program, also stored in memory 3115, which would execute in CPU 3116 to time stamp the data and return a time stamp

certificate to the resident program, which could cause the time stamp certificate to be forwarded to the I/O device 3140 for the user.

Therefore, according to the present invention, key pairs are generated for particular time intervals and time stamp requests are automatically carried out using the private key for the time interval, the private key being destroyed after the time interval. In another embodiment of the present invention, the private key of a prior time interval is used to sign the public key for a subsequent time interval before the private key of the prior time interval is destroyed. In this embodiment of the present invention, every time interval has its own key pair for which the private key is destroyed after signing the public key for the next time interval. According to the present invention, key pairs do not have to be continuously generated every time interval, but can be pre-generated and selected from a queue for each time interval that a time stamp request is received.

The time stamping method according to an embodiment of the present invention uses public key cryptography in a new way to, first, create key pairs that correspond not to fixed entities, such as previous systems employ, but which correspond to transient time intervals; and second, to provide a mechanism to use the keys, and signatures created by those keys, to provide rigorous proof of the time of existence and the authenticity of the content within data signed by the system. As mentioned above, a feature of the system is that the secret key for a given time interval only exists for a finite, typically very short, period of time, and is replaced by subsequent secret keys as subsequent time intervals proceed. A public key cryptography system, such as PGP with

the above-described modifications, is employed to automatically generate a series of public-key encryption key pairs at regular time intervals. Each key contains a designation, for example typically within the key's user ID, which identifies the specific time interval during which it is to be (or was) used. For dynamically-created keys, the minimum possible duration of a time interval is limited by the time necessary for creation of a key pair and the use of that key pair to validate a public key. As indicated above, shorter time intervals can be enabled by pre-generating the key pairs.

As illustrated above, the veracity of the time designation is proven by "chaining" of signatures, so that each new time interval's public key is certified (e.g., digitally signed) using the prior interval's secret key, immediately prior to deleting that prior time interval's secret key. This is done, for example, by using the prior time interval's secret key to digitally sign the new time interval's public key. Immediately after the public key is signed, the prior interval's secret key is deleted

The public key of each key pair is stored for future use. Any given private key is used for time-stamping data only during the time interval immediately following the interval within which the private key was generated. During its interval of use, the secret key is used to digitally sign and time-stamp all data submitted to the system for such processing. As data is submitted to the system for time-stamping, these data are processed by signing them using the respective time interval's private key. This signing process generates a time-stamp certificate. Each time-stamp certificate includes, for example, the digital signature of the data generated by the

secret key and the certified public key for the current time interval of use. Each interval-of-use's public key can be also archived for future reference. For use in easy authentication of time-stamp certificates in the future, all time-stamp certificates can be archived as well, although such time-stamp certificate archiving is not necessary for later proof of the veracity of time-stamps generated by the system.

At the end of each time interval, a new key pair is generated, the public key of the new pair is certified (e.g., signed) by the current time interval's secret key, and that secret key is then deleted, and the cycle continues. Validation of a time-stamp at any later point requires using the respective time interval's public key to authenticate the digital signature in the time-stamp certificate. Validation of that public key is accomplished by using the previous time interval's public key to authenticate the certification signature on the public key to be authenticated. The ability to trace back through the "chain" of public key certification signatures provides irrefutable proof of the location, in time, of any individual time interval's stamp within the chain of signatures. Further evidence of the exact time that a given time interval key was in use can be provided by tracking other certificates that were generated by the same key and collecting evidence of the time of generation of those signatures and the signed data relating to them. Since the secret key for each time interval is destroyed immediately after that time interval passes, it is virtually impossible to create a bogus time-stamp after the fact.

Many other implementations of the time stamping method according to an embodiment of the present invention are possible as well. As described above, for example, one could

calculate the message digests at the users' sites, and send only those message digests to the server for signing. This would both insure confidentiality of data and efficient network bandwidth usage.

WHAT IS CLAIMED IS:

1. A method for certifying data, comprising the steps of:

generating a key pair at a first time interval, the key pair including a private key and a public key;

receiving an certification request;

automatically responding to the certification request by digitally signing data associated with the certification request using the private key; and

deleting the private key.

2. The method according to claim 1, further comprising the step of generating a time stamp certificate confirming the digital signing of the data.

3. The method according to claim 1, further comprising the step of archiving the public key of the first time interval.

4. The method according to claim 1, further comprising the step of authenticating the digitally signed data using the public key.

5. The method according to claim 1, further comprising the step of determining if a further certification request is received during the first time interval.

6. The method according to claim 5, further comprising the step of, for the further certification request, automatically

responding to the further certification request by digitally signing data associated with the further certification request using the private key, wherein the step of deleting the private key is performed after the further certification request has been responded to.

7. The method according to claim 1, further comprising the steps of:

generating a key pair at a next time interval, the key pair including a private key and a public key;

receiving a next certification request;

automatically responding to the next certification request by digitally signing data associated with the next certification request using the private key of the next time interval; and

deleting the private key for the next time interval.

8. A method for certifying data, comprising the steps of:

generating a first key pair at a first time interval, the first key pair including a first public key and a first private key;

generating a second key pair at a second time interval, the second key pair including a second public key and a second private key;

signing the second public key using the first private key;

deleting the first private key;

processing an certification request during the

second time interval using the second private key; and

deleting the second private key.

9. The method according to claim 8, further comprising the step of archiving the first public key.

10. The method according to claim 8, wherein the step of processing the certification request includes automatically responding to the certification request by digitally signing data associated with the certification request using the second private key.

11. The method according to claim 10, further comprising the step of generating an time stamp certificate confirming the digital signing of the data.

12. The method according to claim 11, wherein the time stamp certificate includes the digital signature and the second public key.

13. The method according to claim 12, wherein the time stamp certificate further includes the first public key.

14. The method according to claim 8, further comprising the step of certifying the digitally signed data using the first public key.

15. A system for certifying data, comprising:

a general purpose computer; and

an I/O device coupled to the general purpose computer, wherein the general purpose computer includes a memory containing a program executable by the general purpose computer, the executable program instructing the general purpose computer to

generate a key pair at a first time interval, the key pair including a private key and a public key,

receive an certification request,

automatically respond to the certification request by digitally signing data associated with the certification request using the private key, and

delete the private key.

16. The system according to claim 14, wherein the general purpose computer has a client-server architecture including a client computer and a server computer.

17. A system for certifying data, comprising:

a general purpose computer; and

an I/O device coupled to the general purpose computer, wherein the general purpose computer includes a memory containing a program executable by the general purpose computer, the executable program instructing the general purpose computer to

generate a first key pair at a first time interval, the first key pair including a first public key and

a first private key,

generate a second key pair at a second time interval, the second key pair including a second public key and a second private key,

sign the second public key using the first private key,

delete the first private key,

process an certification request during the second time interval using the second private key, and

delete the second private key.

18. The system according to claim 16, wherein the general purpose computer has a client-server architecture including a client computer and a server computer.

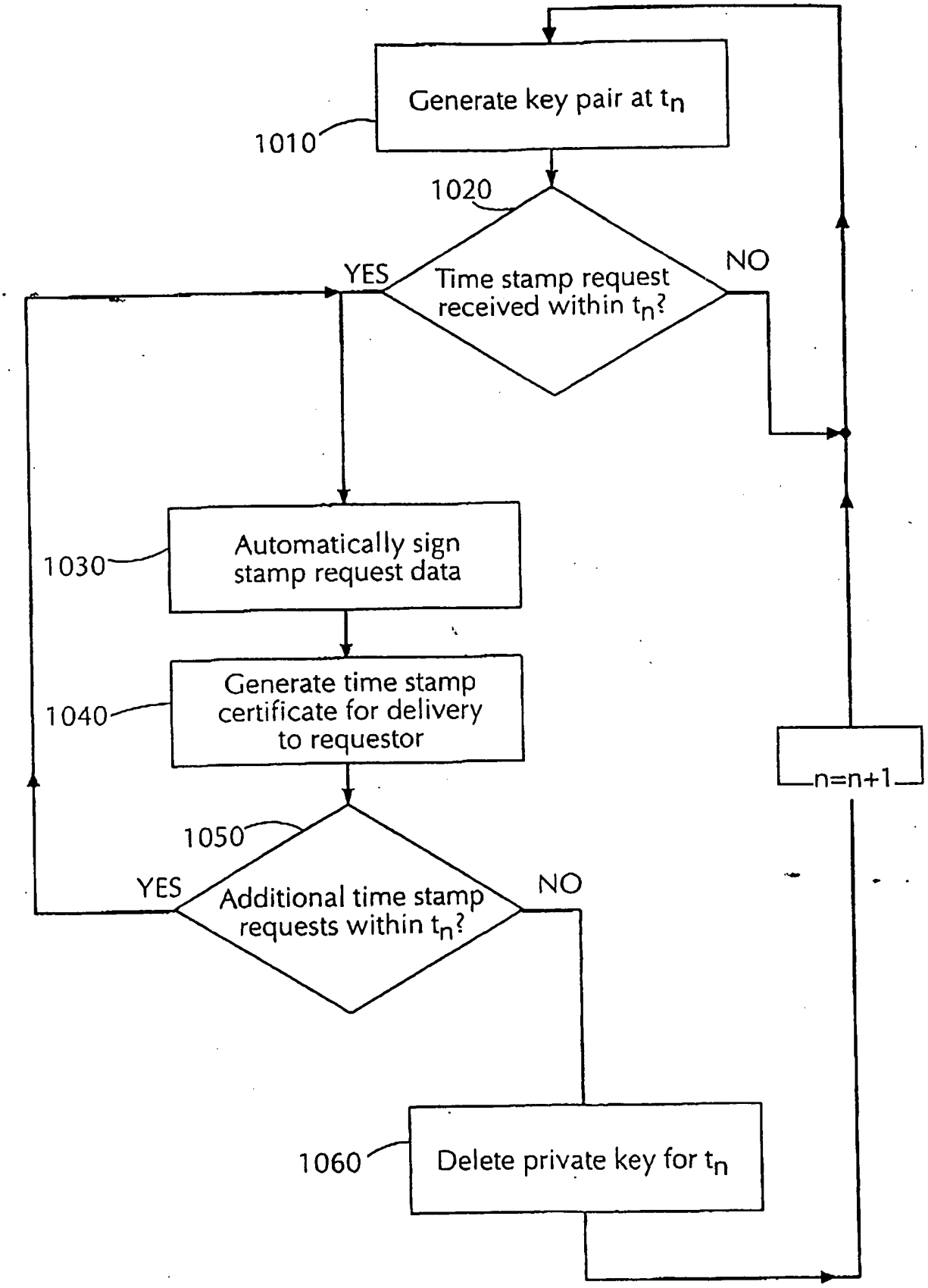


FIG. 1

FIG 2A

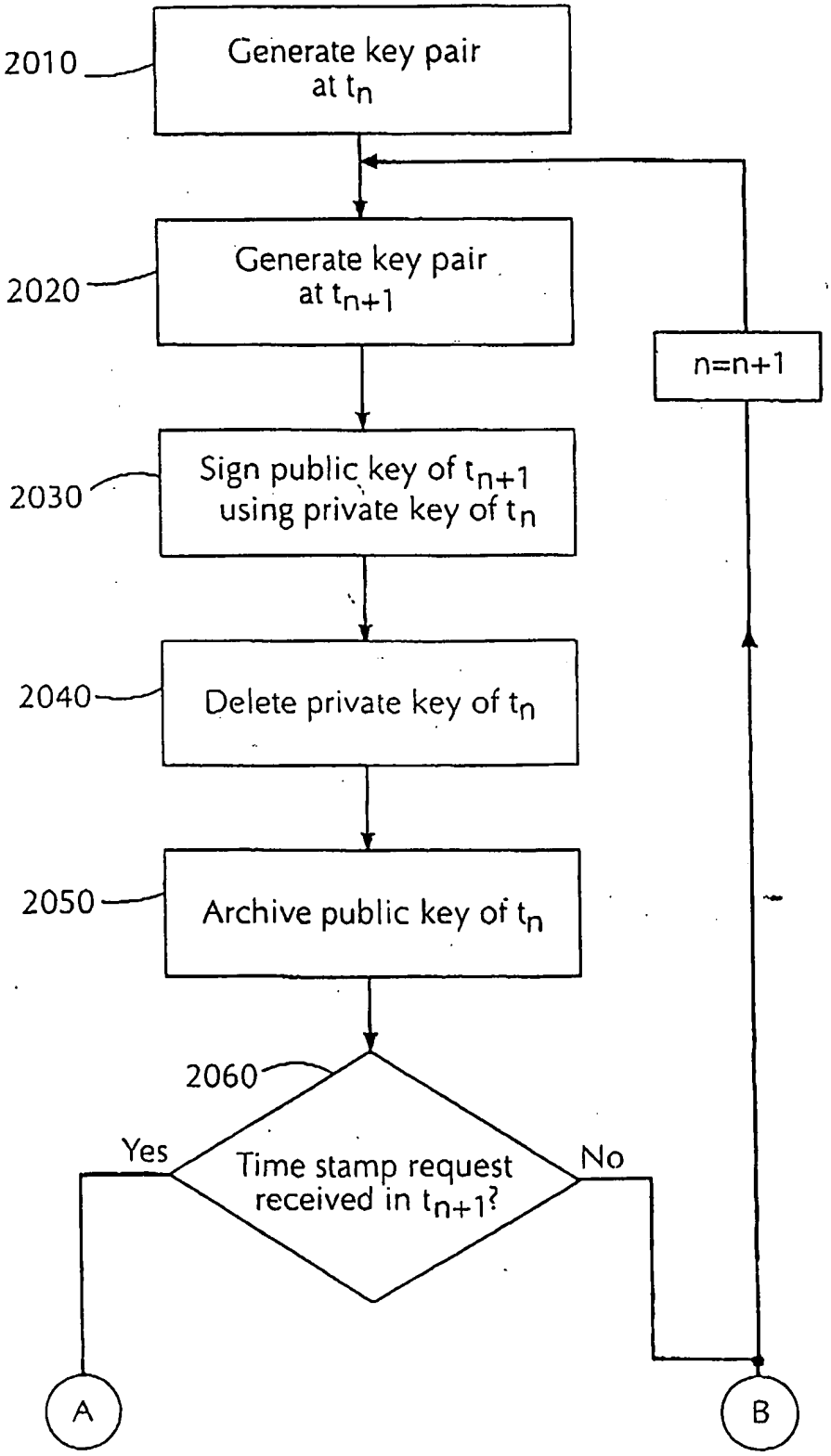


FIG 2B

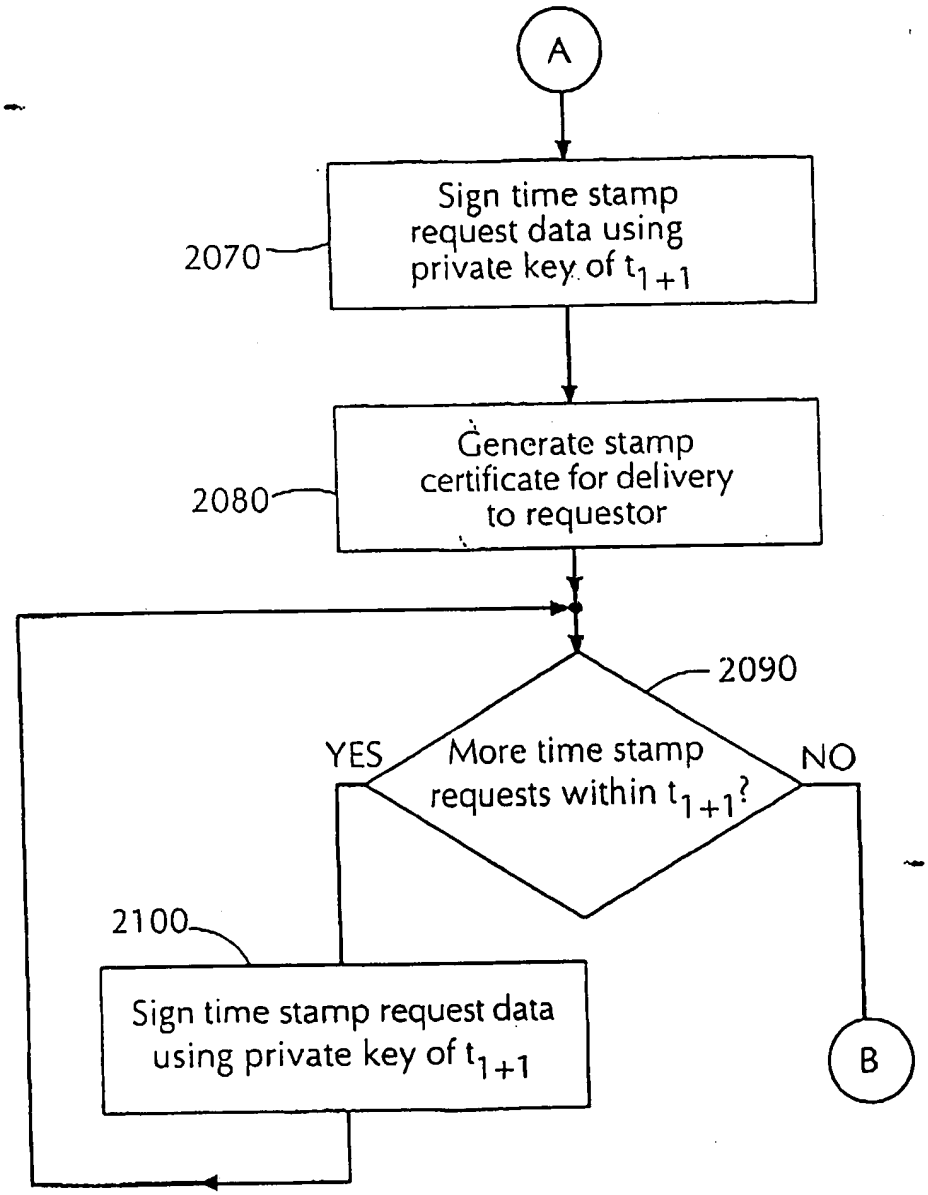


FIG 3A

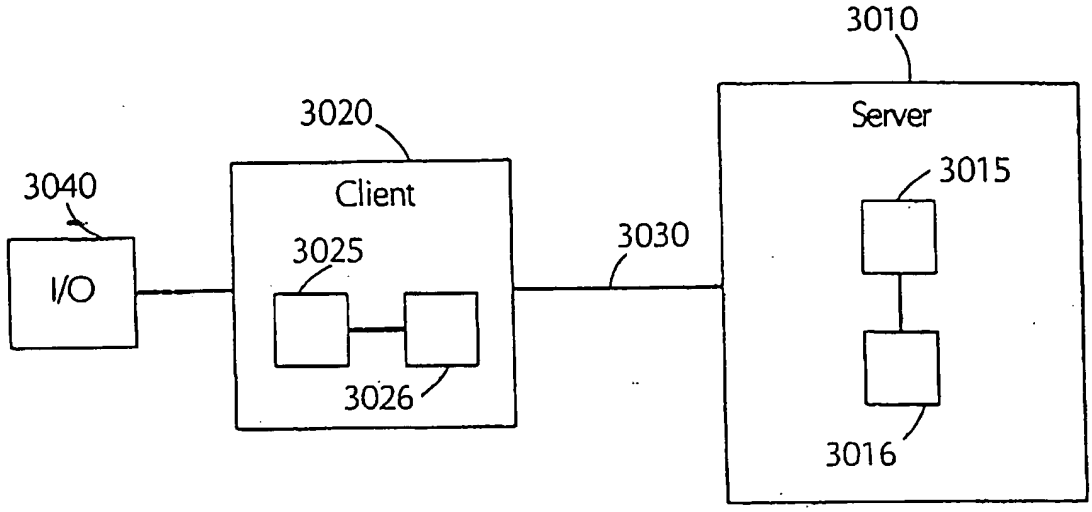
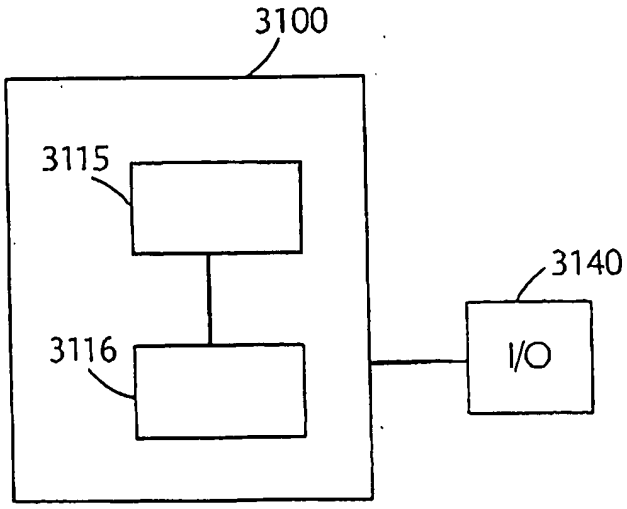


FIG 3B



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/20036

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : H04 L 9/30
US CL : 380/21,30,23

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/21,30,23

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS
search terms:delete, remove,erase,private key, public key, 380/30/ccls, 380/21/ccls

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US 5,787,172 A(Arnold) 28 July 1998 (28.07.98), col. 4,lines 18-47, col. 6, lines 23-38, col.15, lines 19-45, col.17, lines 31-36	1-18
A	US 5,001,752 A(FISHER) 19 March 1991 (19.03.91), abstract, fig.2,4, column 8, lines 12-50.	1-18
A	US 5,469,507 A(Canetti et al.) 21 November 1995 (21.11.95), column 5, lines 55-66, column 6, lines 34-36, fig.3.	1-15, 17
A,P	US 5,673,316 A(AUERBACH et al.) 30 September 1997 (30.09.97), abstract, fig.1, column 7, lines 31-42, column 8, lines 22-25	1-18

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 DECEMBER 1998	26 JAN 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks	Authorized officer

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/20036

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,201,000 A(MATYAS et al.) 04 April 1993 (06.04.93), abstract, column 18, lines 49-66, column 34, lines 27-62, column 36, lines 12-50.	15-18